

BAB II

TINJAUAN PUSTAKA

Penelitian ini tidak terlepas dari hasil penelitian-penelitian terdahulu, jurnal penelitian, artikel di internet serta buku yang berhubungan dengan teknologi web server dinamis dan teori yang berkaitan dengan operating system level virtualisasi menggunakan docker dan kubernetes.

2.1 Availability

Availability atau ketersediaan artinya dalam konteks keamanan informasi upaya untuk menjaga agar sebuah sistem tetap bisa digunakan adalah hal penting yang perlu dilakukan. Kelangsungan sebuah data akan sangat bergantung pada pemeliharaan performa perangkat keras, perangkat lunak, dan saluran komunikasi yang digunakan untuk menyimpan dan memproses informasi. Ketika sebuah aplikasi terganggu dan tidak dapat diakses, maka dapat kehilangan banyak kepercayaan.

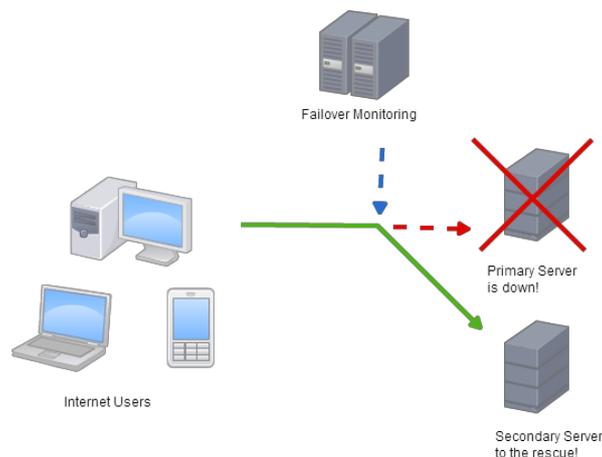
Menurut (Wendy Torell, 2004) mengatakan *“Assures that systems work promptly and service is not denied to authorized users”* yang artinya memastikan bahwa sistem dapat bekerja dengan segera dan layanan tidak ditolak untuk pengguna yang berwenang.

Dikutip dari referensi lainnya (Deepali Mittal, 2015) *“Availability is the guarantee that information will be available to the consumer in a timely and uninterrupted manner when it is needed regardless of location of the user. This means that the cloud infrastructure, the security controls, and the networks connecting the clients and the cloud infrastructure should always be functioning correctly. Availability is ensured by: fault tolerance, authentication and network security”* artinya adalah jaminan bahwa informasi akan tersedia bagi konsumen secara tepat waktu dan tanpa gangguan saat dibutuhkan terlepas dari lokasi pengguna. Ini berarti bahwa infrastruktur, kontrol keamanan, dan jaringan yang menghubungkan klien dan infrastruktur server harus selalu berfungsi dengan benar. Ketersediaan dipastikan dengan : toleransi kesalahan, otentikasi dan keamanan jaringan.

Berdasarkan pemaparan sumber di atas dapat disimpulkan bahwa layanan atau sistem dapat diakses tanpa penghentian atau gangguan terhadap sistem itu sendiri.

2.2 Fail Over

Teknik failover yaitu teknik untuk mencadangkan kinerja salah satu server yang terhubung. Implementasi teknik failover clustering pada server dapat mengurangi kegagalan akses pada server yang dilakukan oleh user. Berikut ini adalah gambaran teknik failover :



Gambar 2. 1 Teknik failover

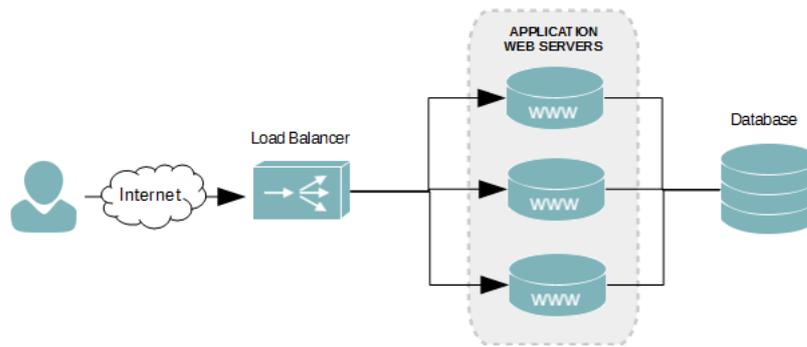
Sumber (Serverfault, 2020)

Menurut penelitian terdahulu (Hakim A, 2018) ketika sistem server gagal terdapat server lainnya yang melakukan redundansi atau cadangan. Dengan menggunakan teknik failover clustering, sistem dapat mendeteksi apakah server utama sedang down, dan secara otomatis akan mentransfer peran server utama ke server cadangan. Dalam penerapan ini, server utama dan server cadangan akan ditempatkan di pusat data yang berbeda.

2.3 Load Balancing

Load Balancing adalah salah satu metode untuk mendistribusikan traffic ke beberapa server agar tidak membebani satu server saja. Alhasil, kekuatan server menjadi seimbang sehingga tidak terjadi overload.

Berikut ini adalah gambaran cara kerja load balancing :



Gambar 2. 2 Cara Kerja Load Balancing

Sumber (Wikipedia, Load Balancing, 2020)

Cara kerja load balancing saat server atau aplikasi Anda menerima traffic dari luar, load balancer tool akan membagikan traffic tersebut ke beberapa server yang tersedia secara rata dan optimal. Berikut ini adalah gambaran dari cara kerja load balancing.

Berikut ini beberapa algoritma load balancing antara lain :

1. *Round Robin*, Sistem load balancer yang paling simple dan paling sering digunakan oleh banyak perangkat load balancer. Beban diberikan secara bergiliran dan berurutan dari satu server ke server lainnya sehingga membentuk sebuah rotation atau putaran.
2. *Ratio*, Algoritma yang satu ini membagikan beban server berdasarkan oleh ratio yang diberikan kepada server-server yang ada. Semakin besar rasio yang dimiliki server, semakin besar beban yang akan diberikan kepada server tersebut.
3. *Fastest*, Sesuai dengan namanya, algoritma yang satu ini membagikan beban server berdasarkan dari the fastest server atau server dengan jaringan yang memiliki respond paling cepat terhadap request dari perangkat load balancer.
4. *Least connection*, Beban server dari algoritma yang satu ini dibagi berdasarkan server yang mempunyai the least connection atau server yang mempunyai koneksi paling sedikit di daftar server-server yang ada.

Berikut ini adalah manfaat penggunaan load balancing antara lain :

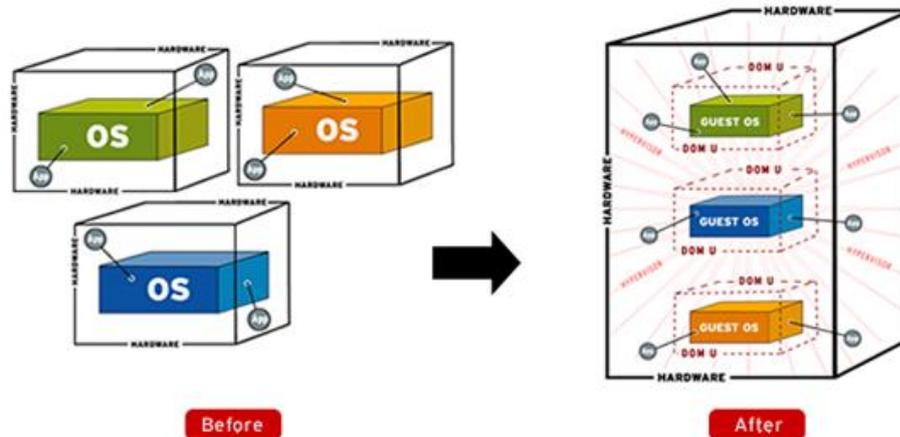
1. *High performance guarantee*, Karena beban traffic dari server dan aplikasi akan dibagi ke beberapa resources atau sumber daya, sudah bisa dipastikan bahwa performa dari server atau aplikasi akan tetap tinggi apapun yang terjadi.
2. *Minimize downtime*, Dengan high performance dan pembagian beban antar server, sudah bisa dipastikan bahwa aplikasi dan situs yang ada di server akan lebih aman dari downtime.
3. *Save resources*, Pembagian beban juga akan membantu Anda menghemat server resources, server tidak perlu menggunakan banyak sumber dayanya untuk tetap menjalankan aplikasi ketika traffic sedang ramai.
4. *More Efficient*, Dengan pembagian beban traffic yang rata akan mendapatkan respons yang lebih cepat dari aplikasi dan server. Hal ini meningkatkan efisiensi dari bisnis.

Penelitian terdahulu yang dilakukan (Moch. Wahyu Imam Santosa, 2018) horizontal scalling dilakukan dengan menambahkan unit pemrosesan seperti node, instance ataupun kontainer merupakan sebuah solusi dalam meningkatkan kinerja sebuah sistem dalam menangani banyaknya permintaan. Load balancing berjalan menggunakan fitur load balancer dan nodeport yang disediakan oleh kubernetes. Dengan terhubungnya setiap replika pada sebuah sistem penyimpanan terpusat menjadikan data bersifat konsisten.

2.4 Virtualisasi

Virtualisasi adalah sebuah teknik yang saat ini banyak diterapkan untuk memenuhi kebutuhan TI yang semakin tinggi namun diikuti dengan tuntutan untuk mengefisiensikan biaya yang digunakan semaksimal mungkin. Virtualisasi adalah teknologi yang telah diterapkan secara luas saat ini dengan dampak peningkatan operasional dan finansial yang positif. Virtualisasi adalah konsep dimana akses ke sebuah hardware seperti server diatur sehingga beberapa operating system (*guest operation system*) dapat berbagi sebuah hardware. Tujuan dari virtualisasi adalah kinerja tingkat tinggi, ketersediaan, kehandalan, atau untuk membuat dasar keamanan dan manajemen yang terpadu.

Berikut ini adalah rancangan dari virtualisasi :



Gambar 2. 3 Virtualisasi

Sumber (Wikipedia, Virtualisasi, 2021)

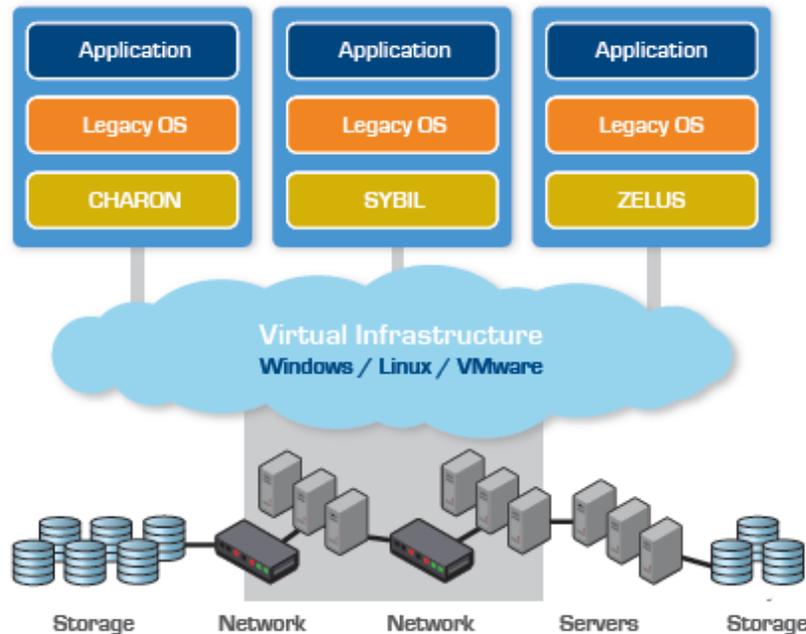
Menurut (Khasanah, 2019) pengembangan teknologi virtualisasi server dengan memanfaatkan sumber daya komputer yang dimiliki oleh instansi menjadi salah satu solusi untuk mengatasi permasalahan tersebut dalam menekan biaya yang dikeluarkan perusahaan untuk pembelian server fisik dan juga meningkatkan performance server.

2.5 Virtual Machine

Virtual machine atau dikenal dengan mesin virtual merupakan sebuah rekayasa perangkat lunak yang dapat berperan seperti komputer fisik. Peran yang diambil seperti menjalankan tugas aplikasi, program, pengelolaan, pengolahan yang sifatnya terselesaikan dengan komputersasi. Apa yang dibangun oleh mesin virtual turut menyertakan fungsi perangkat yang di virtualisasikan. Seperti *log*, *memory*, *file disk*, dan konfigurasi lainnya dibuat dengan rekayasa. Layaknya terdapat rumah di dalam rumah adalah gambaran dari implementasi teknologi ini.

Gambaran sederhana dari implementasi teknologi ini adalah membuat ruang kedua pada komputer, dimana segala aktivitas di dalamnya tidak akan mempengaruhi keadaan komputer utama. Hal ini sangat cocok untuk uji coba program komputer seperti aplikasi atau pembaharuan sistem operasi seperti *windows* dan berguna membuka file yang berisiko terinfeksi virus.

Berikut ini adalah gambaran dari virtual machine (VM) :



Gambar 2. 4 Virtual Machine

Sumber (Wikipedia, Virtual Machine, 2021)

Menurut penelitian terdahulu lainnya (Irwan, 2017) Server yang memiliki kapasitas prosesor tinggi dan kemampuan teknologi virtualisasi umumnya belum sepenuhnya dimanfaatkan secara optimal karena hanya digunakan untuk menyediakan satu layanan aplikasi. Virtualisasi merupakan sebuah teknologi yang memiliki kemampuan dalam memberdayakan sumber daya fisik sebuah server untuk dibangun menjadi beberapa mesin server secara virtual.

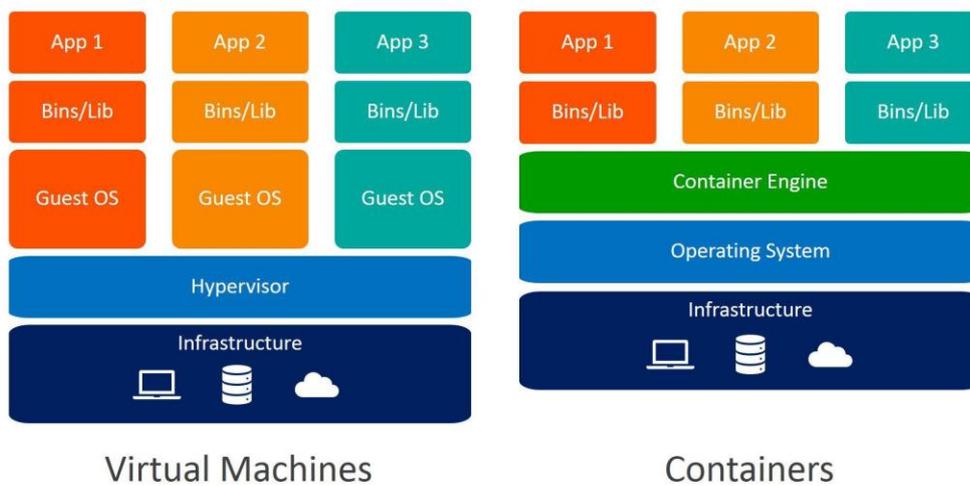
2.6 Container

Container adalah paket atau aplikasi yang mengandalkan isolasi virtual untuk menjalankan aplikasi yang dapat menjalankan sistem operasi kernel secara simultan tanpa memerlukan mesin virtual (VMs). Jika kita flashback kembali ke tahun 60–70an. Proses isolasi virtual sudah ada dan dikembangkan sebagai bagian dari linux. Saat itu isolasi proses dilakukan pada chroot. Bentuk modernnya saat ini diekspresikan dalam bentuk containerization, seperti pada aplikasi docker.

Bagaimana cara kerja container ?. Container menyediakan fleksibilitas secara keseluruhan dibanding kita menggunakan server secara fisik dan virtual

machine. Container dapat langsung berjalan diatas Sistem Operasi (OS) tanpa menggunakan hypervisor. Seperti pada gambar berikut, container berjalan secara langsung diatas Sistem Operasi (OS). Container dapat membagi resource pada OS beserta fungsinya. Hal ini sangat menguntungkan secara performa dan penggunaan resource dikarenakan aplikasi yang digunakan hanya menggunakan resource berdasarkan apa yang dipakai saja.

Apa perbedaan container dengan virtual machine?. Kedua teknologi ini berjalan diatas virtualisasi. Namun, antara container dan virtual machine memiliki perbedaan yang mendasar. Apa saja perbedaannya ?



Gambar 2. 5 Perbandingan VM dan Container

Sumber (Wikipedia, Virtual Machine, 2021)

Berikut ini adalah table perbandingan antara VM dan Container diukur berdasarkan beberapa parameter :

Tabel 2. 1 Perbandingan VM dan Container

No	Parameter	Virtual Machine	Container
1	Guest OS	Setiap VM berjalan pada hardware dan kernel virtual yang dijalankan pada ranah memori virtual	Image kernel dialokasikan pada memori fisik.Semua guest saling berbagi resource yang sama

Tabel 2. 1 Perbandingan VM dan Container

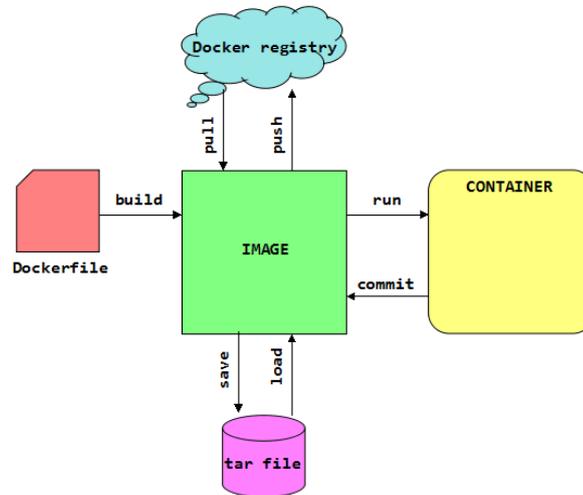
No	Parameter	Virtual Machine	Container
2	Komunikasi	Melalui perangkat hardware internet seperti Ethernet	melalui pipe,signal,socket
3	Keamanan	Bergantung pada hypervisor	memanfaatkan akses yang dibutuhkan
4	Kinerja	Mungkin terjadi sedikit overheat dikarenakan perpindahan Guest OS ke host	Mempunyai kinerja seperti aslinya seperti host OS
5	Isolasi	Tidak bisa berbagi library antar Guest serta Guest dan Host	Subdirectori dipakai dan saling berbagi
6	Waktu Startup	Diperlukan sekian menit untuk melakukan proses booting	Diperlukan sekian detik untuk proses booting
7	Storage	Storage besar karena semua program diinstall dan dijalankan pada satu tempat	Storage lebih kecil karena base OS sharing dan saling berbagi

2.7 Docker

Docker adalah salah satu platform yang dibangun berdasarkan teknologi container. Docker merupakan sebuah project *open-source* yang menyediakan platform terbuka untuk developer maupun *sysadmin* untuk dapat membangun, mengemas, dan menjalankan aplikasi dimanapun sebagai sebuah wadah (container) yang ringan. Dengan sangat populernya docker, sebagian orang sering menganggap docker adalah sebutan lain untuk container.

Dikutip dari situs resmi (Docker, 2020), pengembang dapat mengefektifkan waktu mereka dengan menghilangkan proses konfigurasi yang cocok dengan programnya. Selain itu, berkat fitur sandbox, pengembang leluasa untuk berkreasi tanpa takut merusak programnya. Terakhir docker menjamin program yang kita buat, akan selamanya berjalan seperti seharusnya. Pemaketan

aplikasi dan seluruh kebutuhannya, memastikan aplikasi berjalan lancar pada kondisi lingkungan apapun. Alur kerja docker bisa dilihat di gambar berikut ini :



Gambar 2. 6 Alur kerja Docker

Sumber (Wikipedia, 2021)

Docker adalah sebuah aplikasi yang berbasis teknologi open source yang memungkinkan developer atau siapapun untuk membuat, menjalankan, melakukan percobaan dan meluncurkan aplikasi di dalam sebuah container. Docker membuat proses pemaketan aplikasi bersama komponennya secara cepat dalam sebuah container yang terisolasi, sehingga dapat dijalankan dalam infrastruktur lokal tanpa melakukan perubahan konfigurasi pada container.

Menurut penelitian selanjutnya (Dwiyatno, 2020) docker juga sangat ringan dan cepat jika dibandingkan dengan mesin virtual yang berbasis *hypervisor*. Oleh karena itu, dilakukan implementasi virtualisasi berbasis docker container supaya dapat meningkatkan efektifitas dalam penggunaan sumber daya CPU dan memori pada server.

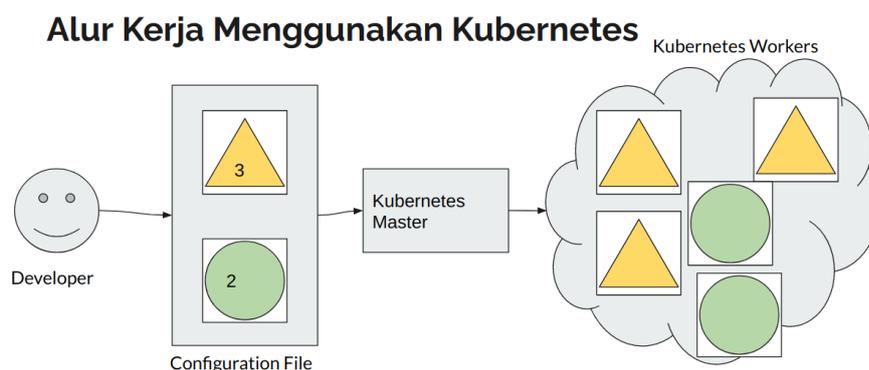
2.8 Kubernetes

Perkembangan teknologi semakin cepat, termasuk datangnya teknologi baru yaitu container. Dengan menggunakan container, seluruh aplikasi yang kita miliki akan di kemas sehingga ketika akan di jalankan dimanapun, aplikasi tersebut akan berjalan sama seperti pada saat kita mencobanya. Kita tidak perlu lagi menginstall server dengan berbagai macam aplikasi, *environment*, dan

kebutuhan lainnya, karena semua yang dibutuhkan oleh aplikasi tersebut sudah terpasang di container.

Untuk menjalankan container, tentunya membutuhkan sebuah aplikasi. Aplikasi yang paling umum digunakan adalah docker. Docker diinstall di server, kemudian kita jalankan *docker image* dan terbentuklah container dan aplikasi kita sudah berjalan dan bisa digunakan. Tetapi untuk kebutuhan *production*, tidak cukup hanya sekedar aplikasi bisa berjalan, kita membutuhkan sistem yang handal yang mampu membuat aplikasi kita selalu available, tidak ada downtime, memiliki sekuriti yang handal, bisa menerima banyak trafik, dan sebagainya. Untuk itu diperlukan lah sebuah orkestrasi (*orchestration*) yang bertugas untuk memamanajemen itu semua. Maka yang bisa kita gunakan untuk itu semua adalah kubernetes.

Kubernetes dikembangkan oleh google dan kubernetes merupakan open source. Semua yang dijalankan di kubernetes, bisa anda simpan dalam file YML, sehingga apapun yang ada didalam *cluster* kubernetes anda, anda akan memiliki dokumentasinya. Sebagai contoh ketika anda ingin membuat *deployment* yang didalamnya terdapat beberapa pod dengan spesifikasi container image nginx:latest, kemudian pod tersebut di ekpose menggunakan service, maka anda hanya perlu membuat sebuah file *deployment.yml*, kemudian anda ketikkan `kubectl create -f deployment.yml` dalam waktu yang singkat apa yang anda butuhkan sudah terbentuk. Berikut ini adalah alur dari kerja kubernetes :



Gambar 2. 7 Alur kerja kubernetes

Sumber (Kubernetes, Kubernetes Concepts, 2020)

Dalam cluster kubernetes terdapat beberapa node. *Node* merupakan server, baik server fisik atau Virtual Private Server (VPS) yang digunakan untuk

menjalankan kubernetes. Untuk membuat cluster kubernetes dibutuhkan setidaknya 1 server untuk kubernetes-master atau node-master dan 1 server untuk kubernetes-node atau node-1.

Kubernetes master adalah server yang bertindak sebagai node-master. Kubernetes master menjalankan tiga komponen yaitu kube-apiserver, kube-controller-manager, dan kube-scheduler.

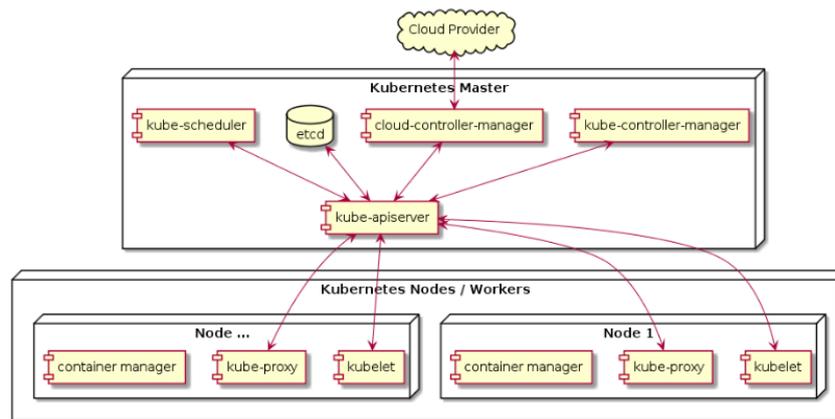
Komponen Master antara lain :

1. *kube-apiserver*, merupakan *front-end* dari kontrol plane Kubernetes dan merupakan komponen di node master yang mengekspos API Kubernetes. Komponen ini didesain agar dapat *di-scale* secara horizontal. Server API Kubernetes memvalidasi dan mengonfigurasi data untuk objek api yang mencakup *pod*, *services*, *replication controller*, dan lainnya.
2. *kube-controller-manager*, merupakan komponen di node master yang menjalankan kontroler. Secara logis, setiap kontroler adalah sebuah proses yang berbeda, tetapi untuk mengurangi kompleksitas, kontroler-kontroler ini dikompilasi menjadi sebuah binary yang dijalankan sebagai satu proses.
3. *kube-scheduler*, merupakan komponen di master yang bertugas mengamati pod yang baru dibuat dan belum di-assign ke suatu node dan kemudian akan memilih sebuah node dimana pod baru tersebut akan dijalankan.
4. Kubernetes nodes adalah server yang terdaftar di server master.

Komponen Nodes antara lain :

- a. *kubelet*, merupakan agen yang dijalankan pada setiap node di kluster dan bertugas memastikan kontainer dijalankan di dalam pod.
- b. *kube-proxy*, bertugas membantu abstraksi service Kubernetes melakukan tugasnya. Hal ini terjadi dengan cara memelihara aturan-aturan jaringan (*network rules*) serta meneruskan koneksi yang ditujukan pada suatu host.

Berikut ini adalah sebuah gambaran arsitektur di dalam kubernetes :



Gambar 2. 8 Arsitektur Kubernetes

Sumber (Kubernetes, Kubernetes Concepts, 2020)

Perkembangan teknologi yang semakin maju menyebabkan banyaknya tuntutan pada client- server untuk menjalankan tugasnya. Kemampuan server yang ada akan semakin menurun dengan banyaknya akses yang datang dari client. Kemampuan server yang melebihi batas akan mengalami overload sehingga banyak dampak buruk yang akan ditimbulkan antara lain penurunan kecepatan waktu mengakses bahkan menimbulkan server mati.

Menurut penelitian lainnya selanjutnya (Kezia Yedutun, 2020) teknologi containerization ini diimplementasikan dengan memanfaatkan scalability. Scalability pada container akan menyesuaikan dengan kebutuhan client maupun server. Dengan teknologi container, akses terhadap server dapat dilakukan dengan penambahan container untuk jalur akses jika dirasa penuh dan akan mengurangi container jika jumlah client yang mengakses berkurang. Hal ini akan meminimalisir server yang overload sehingga client dapat melakukan akses terhadap server tanpa kesulitan akibat sedikitnya jalur akses yang ada.

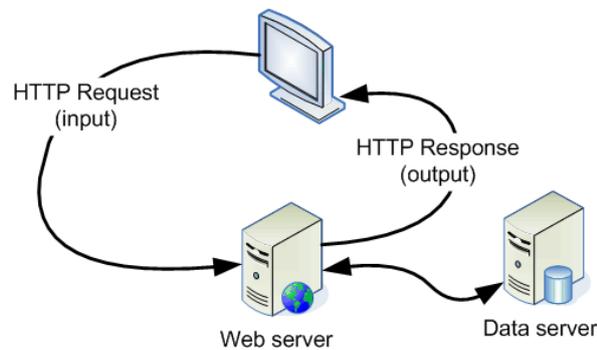
2.9 Web Server

Merupakan perangkat lunak yang merespon permintaan (request) dari browser. Web server mempunyai peran untuk menghubungkan antara user dengan server website. Contohnya ketika user ingin mengakses website abc.com melalui browser. Pada saat user memasukkan URL (abc.com) ke dalam browser kemudian mengaksesnya, *browser* mengirimkan request ke web server. Kemudian web server mencari data dari website abc.com . Setelah menemukan data website

abc.com, web server kemudian mengirimkan HTTP response ke browser dan mengubahnya menjadi halaman website.

Web server berbeda dengan file server, mail server, atau database server. Maka dari itu, masing-masing dari jenis server ini menggunakan perangkat lunak server yang berbeda-beda. Aplikasi file server contohnya seperti Samba dan Google Drive (Online). *Sendmail* dan *Postfix* merupakan contoh mail server yang ada saat ini. Sedangkan contoh dari database server adalah MySQL dan Microsoft SQL Server. Contoh dari perangkat lunak web server yang saat ini adalah *Nginx*, *Apache*, dan *LiteSpeed*. Di antara ketiganya, Apache merupakan web server yang paling banyak digunakan berdampingan dengan *Nginx*.

Berikut ini adalah gambaran dari cara kerja suatu web server :



Gambar 2. 9 Cara kerja web server

Sumber (Wikipedia, Web Server, 2021)

2.10 PHP

PHP Adalah bahasa scripting *server-side*, Bahasa pemrograman yang digunakan untuk mengembangkan situs web statis atau situs web dinamis atau aplikasi Web. PHP singkatan dari Hypertext Pre-processor, yang sebelumnya disebut *Personal Home Pages*. *Script* sendiri merupakan sekumpulan instruksi pemrograman yang ditafsirkan pada saat runtime. Sedangkan Bahasa scripting adalah bahasa yang menafsirkan skrip saat runtime. Dan biasanya tertanam ke dalam lingkungan perangkat lunak lain. Fungsi PHP adalah membuat atau mengembangkan situs web statis atau situs web dinamis atau aplikasi Web. Walaupun sebenarnya bukan hanya PHP bahasa pemrograman yang bisa digunakan untuk memuat website (Wikipedia, PHP, 2021).

2.11 MySQL

MySQL adalah *Database management system*(DBMS) yang open source dengan dua bentuk lisensi, yaitu Free Software (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah database server yang gratis dengan lisensi GNU General Public License (GPL) sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada. SQL sendiri merupakan suatu bahasa yang dipakai di dalam pengambilan data pada relational database atau database yang terstruktur. Jadi MySQL adalah database management system yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan database server (Wikipedia, Mysql, 2021).

Dari sekian sumber yang diperoleh semua bertujuan untuk meningkatkan ketersediaan layanan webserver dan aplikasi. Dapat disimpulkan bahwa, perkembangan teknologi yang semakin maju menyebabkan banyaknya tuntutan pada client- server untuk menjalankan tugasnya. Kemampuan server yang melebihi batas akan mengalami overload sehingga banyak dampak buruk yang akan ditimbulkan antara lain penurunan kecepatan waktu mengakses bahkan menimbulkan server mati. Oleh sebab itu perlu adanya metode yang bisa mengurangi beban server dan meningkatkan availability pada sisi kinerja server. Teknologi containerization salah satu solusinya dari masalah yang ada, dimana container merupakan jalur akses bagi client ke server. Dengan teknologi container, akses terhadap server dapat dilakukan dengan penambahan container untuk jalur akses jika dirasa penuh dan akan mengurangi container jika jumlah client yang mengakses berkurang. Hal ini akan meminimalisir server yang overload sehingga client dapat melakukan akses terhadap server tanpa kesulitan akibat sedikitnya jalur akses yang ada.